Immersing NURBS for CFD applications

Jérémy Veysset, Ghina Jannoun, Elie Hachem, Thierry Coupez

YIC2013
Second ECCOMAS Young Investigators Conference
2–6 September 2013, Bordeaux, France

Immersing NURBS for CFD applications

J. Veysset [a,*], G. Jannoun [a], E. Hachem [a], T. Coupez [a]

[a] Mines ParisTech, Center for Material Forming (CEMEF),
UMR CNRS 7635, BP 207, 06904 Sophia-Antipolis, France

*jeremy.veysset@mines-paristech.fr

Abstract. *We present a new immersed method for solving conjugate heat transfer and fluid-solid interactions (FSI). It is based on the use of Non Uniform Rational B-Splines (NURBS) to compute the distance function ans thus representing the immersed solids inside the computational domain. Combined with anisotropic mesh adaptation and stabilized Finite Elements Method (FEM), it allows a novel, efficient and flexible approach to deal with turbulent flows and heat transfer inside large domains.*

Keywords: NURBS; Anisotropic mesh adaptation; FSI.

1 INTRODUCTION

In the field of numerical computation using FEM, simulating FSI coupled to heat transfer is crucial for many industrials. The challenge and the interest is to be able to predict the thermal history of the objects in their environment (industrial furnaces, quenching device, or even weather forecast...). In order to compute such phenomena, two main phylosophies predominate. The first one consists in representating the objects in a Lagrangian way, generating the mesh of the computational domain that fits all around the objects. The domain and the objects equations are then coupled with exchange coeffients. The second one deals with an Eulerian point of view. The computational domain is represented by a single mesh, and the objects are captured by means of level-sets. This technic is called the Immersed Volume Method (IVM) beacause the objects are immersed in the global mesh. As there is only one set of equations, this method provides an efficient way to solve turbulent flows coupled to conjugate heat transfer around complex geometries.

Usually the distance function is computed relatively to the mesh of the objects. This approach has proved to be generic and provides fast results, but its precision highly depends on the initial mesh files describing the geometry of the objects. Moreover, the generation of these mesh files from the Computer Aided Design (CAD) definition of the object can be really tricky. We propose here a new method to avoid this precision dependancy and to skip the step of the generation of the mesh files. The method consists in immersing the objects in the computational domain directly from their CAD definition by computing their distance function respectively to NURBS curves or surfaces. Since the objects are no more meshed, the method provides an improved accuracy and an original and more flexible way of immersing objects.

To capture and valorize the precision given by this new method, we use anisotropic mesh adaptation. The mesh adaptation creates higly stretched elements that are essential to keep an accurate description of the fluid-solid interface, and track all the physics and scales as well with a reasonable computational time. As the elements are anisotropic, the use of a stabilized FEM is needed.

This paper is organized as followed. In section 2 we present the new immersion approach using NURBS. The method is detailed and an algorithm is provided. Then the mesh adaptation technic is introduced, as well as the solvers that are involved. Finally in section 3, 3D numerical examples and industrial applications are given.

2 IMMERSED NURBS METHOD

2.1 Computing Level-sets from NURBS

The goal here is to compute the level-set (the signed distance function) of the objects involved in the simulations directly from their CAD definition, i.e. their CAD files. In these CAD files, the objects are commonly caracterized by NURBS curves or surfaces. Let Ω, Ω_f, Ω_s and Γ represent respectively the whole domain, the fluid domain, the solid domain and the interface. They verify:

$$\left\{ \begin{array}{c} \Omega_f \bigcup \Omega_s = \Omega \\ \Omega_f \bigcap \Omega_s = \Gamma \end{array} \right\} \tag{1}$$

Then for each node X of the computational domain Ω, the level-set function α which is the signed distance from the interface reads:

$$\alpha(X) \left\{ \begin{array}{c} > 0 \text{ if } X \in \Omega_s \\ = 0 \text{ if } X \in \Gamma \\ < 0 \text{ if } X \in \Omega_f \end{array} \right\} \tag{2}$$

The physical and thermodynamic properties in the domain are then calculated as a function of α; for instance, the mixed density and viscosity are calculated using a linear interpolation between the values of the fluid and the solid. NURBS functions are used in the CAD field to represent curves and surfaces. With such mathematical functions, it is possible to represent any geometry. These parametric functions are piecewise polynomial. The definition of a NURBS curve c is as follows:

$$c(u) = \frac{\sum_{i=1}^{n} N_{i,p}(u)\omega_i P_i}{\sum_{i=1}^{n} N_{i,p}(u)\omega_i} \tag{3}$$

with $N_{i,p}$ the basis functions, $_i$ the weights, P_i the control points, p the degree of the curve, n the number of control points and u the parameter taking its values in the knot vector U. The definition of a NURBS surface s is similar to the one of the NURBS curve:

$$s(u,v) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(u) N_{j,q}(v)\omega_{i,j} P_{i,j}}{\sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(u) N_{j,q}(v)\omega_{i,j}} \tag{4}$$

with $N_{i,p}$ and $N_{j,q}$ the basis functions, $_{i,j}$ the weights, $P_{i,j}$ the control points, p and q the degree of the surface in the u and v directions, n and m the number of control points in the u and v directions and u and v the parameter taking their values in the knot vectors U and V. For further details on NURBS curves and surfaces, the reader is invited to read [1].

Computing the level-set of any object is equivalent to finding the closest point on the object from any node of the computational mesh. The closest point problem can be seen as a root finding problem [2]. In fact, if we consider a point P and a NURBS curve c, the projection of the point P on the curve c is mathematically equivalent to finding the parameter u^* such that:

$$(P - c(u^*)).c^{'}(u^*) = 0 \tag{5}$$

This kind of problem can be solved by using a Newton method. But this method requires a good initial value in order to get a good result. Thus the methods found in the literature differ in setting the initial value. In [1], the authors make a sampling of points on the curve and take as an initial value the most closed to the curve. But this method has been described as time consuming. A more efficient way to find a good initial value is to select the part of the curve containing the root. Then the initial value is taken on this part of the curve and the Newton method is performed only on this part. In most of the articles, the technique consists in decomposing the curve in several Bezier curves.

Then a criterion is used in order to check which Bezier curve contains the root [3, 4, 6, 5]. The method proposed here is inspired from these articles. First we check if one of the extremities of the curve is the closest point by using the following criteria:

if$\forall i \in [1, n] P_1P_i.PP_1 \geq 0$ then P_1 is the closest point,

P being the test point from which we want to find the closest point on the curve, and P_i the control points of the curves. With the same idea:

if$\forall i \in [1, n] P_nP_i.PP_n \geq 0$ then P_n is the closest point [5].

Otherwise the closest point is not an extremity and the curve is subdivided into Bezier subcurves. Then for each Bezier subcurve B_k, we eliminate all the subcurves B_k whose closest point from point P is an extremity of B_k by using the same criteria [5]. If all the subcurves B_k have been suppressed, then the curve has got at least one singularity and the closest point is one of these singular points (point of multiplicity equal to $p-1$, p being the order of the curve). Thus we compute the distance for all the singular points and check which one is the closest. Otherwise we look for the closest point with a Newton method. Finally, we sign the distance.

2.2 Anisotropic Mesh Adaptation

We apply the theory developped in [7] and construct a metric directly at the mesh nodes. For this purpose, we use the statistical concept of the length distribution tensor. The error is computed along the edges and in the direction of each edge. It is proved that the use of this anisotropic mesh adaptation method allow the creation of extremely stretched elements ans thus ensuring a geometric accurate representation and convergence.

2.3 Navier-Stokes (NS) and Convection-Diffusion-Reaction (CDR) solvers

We solve the NS and the CDR equations using the stabilized FEM developped in [8]. It is well known that the classical finite element approximation for the flow problem may fail because of two reasons: the compatibility condition known by the inf-sup condition or Brezzi-Babuska condition which requires an appropriate pair of the function spaces for the velocity and the pressure and when the convection dominates. Therefore, we employ stable finite element formulation based on the enrichment of the functional spaces with space of bubble functions known as Mini element. The special choice of bubble functions enables us to employ static condensation procedure giving rise to a stabilized formulation for equal-order linear element.

3 NUMERICAL AND INDUSTRIAL APPLICATIONS

3.1 Immersion of a ship hull

The goal here is to test the NURBS immersion method by computing the level-set relatevely to a ship hull. The hull is described only by its CAD representation. It is composed of four NURBS surfaces. We initially immerse the hull into a mesh refined in the region of the object. The initial mesh is around 102,000 nodes (figure 1). Then we adapt the mesh on the computed level-set with the anisotropic mesh adaption method described in subsection 2.2. The final adapted mesh is around 119,000 nodes. We can see on figure 1 the contour of the zero-value of the level-set for the different meshes. The more the mesh is well adapted on the level-set the more the object definition is accurate.

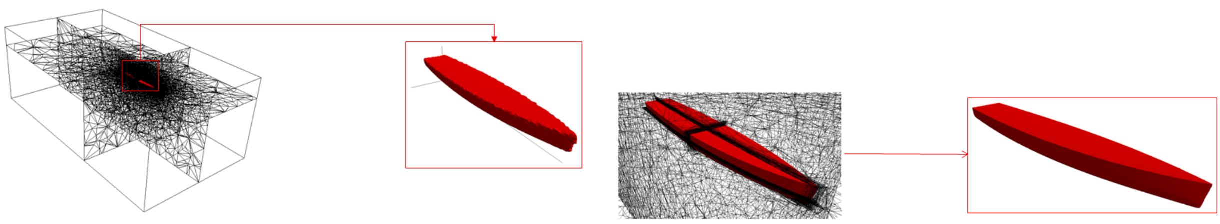

Figure 1: Initial mesh (left): 102,000 nodes, and adapted mesh (right): 119,000 nodes

3.2 Ingots heating in an industrial furnace

In this section, we aim to present the heating process of an industrial furnace given by an industrial partner. Six ingots with arbitrary geometries stand inside the furnace. They are taken initially at 640C and positioned at different locations. Air at 1352C is injected at the input. The mesh is adapted on the level-sets of the ingots as well as on the temperature field, using the anisotropic mesh adaption method described in subsection 2.2. NS and CDR equations are solved by the stabilized FEM introduced in subsection 2.3. The mesh was adapted before starting the simulation on the level-sets of the ingots. We can notice on figure 2 how their geometry is well captured and how the elements of the mesh are concentrated and highly stretched all around their interfaces. On figure 2, we can notice how the mesh follows the ingots level-sets as well as the temperature field.

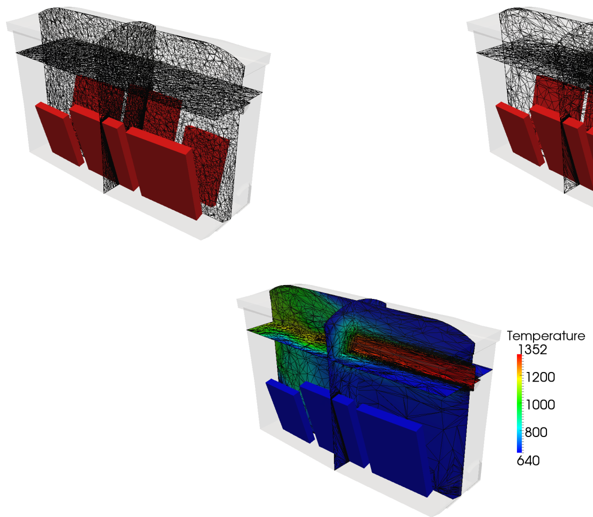

Figure 2: Initial setup of the furnace (top-left, mesh pre-adapted on the ingots level-sets, 990,000 elements), Adapted mesh (otp-right) and temperature field (bottom) in the furnace during the simulation, 920,000 elements

4 CONCLUSIONS

We have shown a new method to immerse the objects directly from their CAD definition. The objects are no more described by mesh files, but by CAD files. The immersion is done by computing the level-sets of the objects respectively to the NURBS curves and surfaces describing them. Then the mesh is adapted on the level-sets to capture accurately their geometries. The method provide a flexible and novel approach to deal with complex geometries. 3D Examples and industrial applications have been provided, through the immersion of ship hull and the heating of an industrial furnace containing six ingots.

REFERENCES

[1] L.A. Piegl and W. Tiller. *The NURBS Book 2nd Edition*. 1996.
[2] P. Schneider and D. Eberly. *Geometric Tools for Computer Graphics*. 2003.
[3] E. Dyllong and W. Luther. Distance calculation between a point and a nurbs surface. *Curve and Surface Design*, Saint-Malo, 55-62, 1999.
[4] Y.L. Ma and W.T. Hewitt. Point inversion and projection for nurbs curve and surface: Control polygon approach. *Computer Aided Geometric Design*, **20**:7999, 2003.
[5] E. Cohen and D. Johnson. Distance extrema for spline models using tangent cones. *Proceedings of the Graphics Interface 2005 Conference*, Victoria, British Columbia, Canada, 169175, 2005.
[6] I. Selimovic. Improved algorithms for the projection of points on nurbs curves and surfaces. *Computer Aided Geometric Design*, **23**:439445, 2006.
[7] T. Coupez, Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing, *Journal of Computational Physics* **230**:2391-2405, 2011.
[8] E. Hachem. *Stabilized Finite Element Method for Heat Transfer and Turbulent Flows inside Industrial Furnaces*. PhD thesis, Ecole Nationale Suprieure des Mines de Paris, 2009.